Public reporting burden for this collection of information is estimated to average 1 hour per response, including the ti gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comm collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Direc Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		1 June 1996 - 31 Dec 99

4. TITLE AND SUBTITLE
Fast (Parallel) Algorithms for Spherical Transforms and Many-Body Interactions with Applications in Electrostratics, Image Processing and Chemistry

5. FUNDING NUMBERS
F49620-96-1-0289

6. AUTHOR(S)
S. Lennart Johnsson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Computer Sciences
University of Houston
4800 Calhoun Blvd.
Houston, TX 77204-2163

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR
801 N. Randolph Street, Room 732
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-96-1-0289

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release.

12b. DISTRIBUTION CODE

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
NOTICE OF TRANSMITTAL DTIC. THIS TECHNICAL REPORT
HAS BEEN REVIEWED AND IS APPROVED FOR PUBLIC RELEASE
LAW AFR 190-12. DISTRIBUTION IS UNLIMITED.

13. ABSTRACT *(Maximum 200 words)*

The main accomplishments during this grant period are in the following areas:
1. Fast parallel algorithms for Legendre and spherical transforms. The accomplishments in this area effectively break down into two subareas, namely:
(a) Development of fast algorithms for the discrete Legendre trans-form (FLT).
(b) Development of factorizations of the Fourier transform (FFT) and the cosine transform (FCT) adapted for the use in spherical trans-forms.
2. Fast N body algorithms.
3. Efficient parallel codes for electromagnetic field computations.

14. SUBJECT TERMS

15. NUMBER OF PAGES
10

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

FAST (PARALLEL) ALGORITHMS FOR SPHERICAL TRANSFORMS AND MANY-BODY INTERACTIONS WITH APPLICATIONS IN ELECTROSTATICS, IMAGE PROCESSING AND CHEMISTRY

GRANT NO F.49620-96-1-0289

Final Report
June 1996 – July 31, 1999

S. Lennart Johnsson and Dragan Mirković
Department of Computer Science
University of Houston
Johnsson@cs.uh.edu, Mirkovic@cs.uh.edu

February 28, 2001

1

1   Introduction

It has been well known for many years that function representations in spherical harmonics is very effective in several fields of science and engineering, such as quantum chemistry, global weather and climate simulations, image processing, control system design and astrophysics to mention a few. Spherical harmonics are eigenfunctions to the Laplace operator in spherical coordinates.

A few such applications occur in chemistry and molecular dynamics. These applications also require that many–body interactions be modeled and evaluated effectively. Anderson's method based on Poisson's formula is a very competitive technique for large particle systems.

The research during this grant period was focused on the development and implementation of fast parallel algorithms for spherical and related Legendre and Fourier transforms, efficient implementations of N–body algorithms and development of efficient parallel codes for electromagnetic field computations.

2   Accomplishments

The main accomplishments during this grant period are in the following areas:

1. Fast parallel algorithms for Legendre and spherical transforms. The accomplishments in this area effectively break down into two subareas, namely:

   (a) Development of fast algorithms for the discrete Legendre transform (FLT).

   (b) Development of factorizations of the Fourier transform (FFT) and the cosine transform (FCT) adapted for the use in spherical transforms.

2. Fast N–body algorithms.

3. Efficient parallel codes for electromagnetic field computations.

2.1 Fast Legendre Transforms

For the fast Legendre transforms of computational complexity $O(Nlog_2^2N)$ we devised a parallel code and investigated its performance on Connection Machine systems. The code is based on an algorithm devised by Driscoll and Healy with subsequent improvements and variations by Healy, Moore and Rockmore [2, 1]. To our knowledge our code is the first and only parallel code of this new algorithm.

The formal algebraic description of algorithms developed as part of this work is suitable for both performance analysis and program transformation techniques. Such manipulation is an integral part of high–level compilation systems. The results of this work has been documented in two technical reports, two conference publications, two journal submissions and one PhD. thesis.

2.2 Fast Fourier Transforms

We have developed an efficient FFT library which contains code that adapts itself to the processor architecture by factoring the Fourier transform at run–time depending upon its size. The library contains a number of composable blocks of code, called modules or codelets, each computing a part of the transform, and initialization/execution routines. Given the parameters of the problem the initialization routine selects the optimal execution strategy in terms of the actual execution time on the given architecture. The modules in the library are highly optimized and generated by a special–purpose compiler that we have developed. The execution routine uses a combination of the mixed-radix splitting and the prime factor algorithm (PFA). The approach is similar to what is used in the FFTW project [3], but differs in optimization and execution strategy.

The overall efficiency of the code depends strongly on the efficiency of the modules, but the actual coding and optimization becomes very tedious and difficult for sizes greater then 5. For this reason many authors have found it convenient to build the modules by using different ways of automatic code generation. Our code generator is written in C and it can produce DFT modules of arbitrary size, direction (forward or inverse), and rotation (for PFA.) It first generates an abstraction of the FFT algorithm by using a combination of Rader's algorithm [6], mixed-radix algorithm and PFA. The next step is the optimization and scheduling of operations. This optimization

is also architecture dependent and it is performed only once, during the installation of the library. Finally, the abstract code is unparsed to produce the desired C code. An important feature of the code generator that we have developed is its generality, hence it can be easily extended to problems other than generation of FFT modules, such as generation of modules for FLTs and FCTs.

The initialization of the code consists of the fast computation of twiddle factors and other constants needed for the transform, and of selecting the optimal factorization of the Fourier transform of a given size. This approach makes the code adaptive to the architecture it is running on. First, we use a combination of the mixed–radix [5] and the prime factor algorithm (PFA) [5] splittings to generate a large number of possible factorizations for a given transform size. Next, we select the fastest factorization in terms of the actual execution time on the given architecture.

We have developed recursive execution routines which are based on the combination of the mixed–radix and the prime factor algorithms.

The results of this work have been documented in one conference publication and two MS theses.

2.3   Fast N–body Algorithms

One major accomplishment is the data parallel partitioning technique implemented in High Performance Fortran (HPF). This software is viewed as a major accomplishment in that it is to our knowledge the first implementation of a partitioning routine for irregular data structures in the data parallel language HPF, and the implementation is efficient.

Our implementation of an adaptive N-body code for three-dimensional problems is also viewed as a breakthrough in that it again achieves high efficiency for nonuniform computations despite its implementation in HPF. The breakthrough lies in the techniques used for handling the nonuniformity efficiently. The techniques apply to other nonuniform problems as well.

Another major accomplishment is a technique for calling HPF programs from MPI programs. Such calling sequences are not defined in HPF, while the converse is part of the HPF definition. HPF has global data structures, and the information about these data structures can be inherited by MPI programs through an interface known as intrinsic procedures. However, there is no global data structure information in an MPI program, and thus calling an HPF program from an MPI code represents a serious challenge.

4

Our techniques, communication with UNIX sockets, communication with MPI and communication using shared memory, have been tested and run on a distributed memory machine (IBM SP2), shared memory machines (HP Exemplar, SGI Origin 2000) and the Globus metacomputing toolkit. We believe our techniques will be useful for integrating existing MPI and HPF codes. In addition, MPI-HPF communication will be useful, where a mixture of data and task parallelism is desired. This is because MPI is popularly used for implementing task parallelism and HPF is an effective language for implementing data parallel applications.

With respect to developing an understanding of the error properties of Anderson's method, and comparing it to other similar methods, such as the Greengard–Rohklin fast multi–pole method (GRFM), we have empirically established that the errors of the two methods is approximately the same for an equal number of terms in the expansions, i.e., the integration order in Anderson's method needs to be about twice that of the multipole order of GRFM for approximately the same error. To achieve high accuracy the integration order in Anderson's method needs to be increased beyond the 14th order for which formulas were provided in the reference Anderson used. We initiated an investigation of higher order integration formulas on the sphere that are computationally efficient.

For the fast N–body codes we made simulations determining the error for Anderson's method as a function of the integration order (and the number of terms in the expansion) for both some synthetic cases as well as for some molecules used in studies at the materials Laboratory at Wright Patterson Air Force Base. The largest molecule had 80,712 atoms. The error was determined as the difference between the potentials computed by Anderson's method and the direct N–body algorithm requiring $O(N^2)$ operations. The same cases were studied using the GRFM code from the materials Laboratory.

To achieve high accuracy the integration order in Anderson's method needs to be increased beyond the 14th order for which formulas were provided in the reference Anderson used. We have reconstructed numerical integration formulas for the surface of the sphere of orders 9, 11, 13, 15, 17, 19, and 23, using the approach given in [4].

The adaptive code has been ported and tested on IBM SP2 at University of Houston, HP Convex Exemplar X-Class at California Institute of Technology and on Globus (IBM SP2) testbed with PGHPF compiler(PG-Portland Group).

5

2.4 Efficient parallel codes for electromagnetic field computations

We have developed a Computational Electromagnetics code based on a high-order algorithm due to Joseph Shang [7] for computing the scattering of an electromagnetic wave from geometrically complex objects. Our parallel code is new (not a port of a sequential code) and it is optimized at the algorithmic and at the (parallel) implementation level. The code is arithmetically more efficient than the original sequential code (it uses about 70% as many arithmetic operations as the original code). It uses an automatic domain partitioning that minimizes the communication needs and loop reordering for minimum looping overhead and maximum efficiency in use of local memory hierarchies.

We have performed a performance analysis which includes estimates of the best possible performance with respect to arithmetic capabilities of the processor architectures and operations in the code, as well as the peak performance with respect to memory bandwidth. On IBM SP platforms, a Fortran 90 with MPI implementation achieves a sequential efficiency of about 70% of estimated peak. Communication accounts for less than 10% of the execution time. The performance is about 40% better than the previous best implementation of the same algorithm on SP systems. The code is highly modular and includes a flexible domain decomposition routine that as a default, distributes the domain evenly across all processors in a way that minimizes the necessary communication. We have also developed a distributed latency–tolerant IBM SP implementation, using the vBNS backbone and the Globus toolkit.

3   Personnel Supported

Principal Investigator: S. Lennart Johnsson
Research Associates: Dragan Mirković and Zdenko Tomašić.
Graduate Students: Yu Charlie Hu, Nadia Shalaby, Olle Larsson, Hrishikesh P. Divate, Anil-Kumar Padala and Priti Mehta.

4   Publications

4.1   Ph.D. theses

1. Nadia Shalaby, "Fast Parallel Orthogonal Transforms", Harvard University, spring 1997.

2. Yu Charlie Hu, "Efficient data parallel implementations of highly irregular problems", Harvard University, spring 1997.

4.2   MS theses

1. Olle Larsson, "Implementation and performance analysis of a high–order CEM algorithm in Parallel and Distributed Computing Environments", University of Houston, May 1998.

2. Hrishikesh P. Divate, "MPI – HPF Communication Techniques", University of Houston, May 2000.

3. Rishad Mahasoom, "An adaptive software library for fast Fourier transforms", University of Houston, December 1999.

4. Fredrick Mwandia, "An adaptive software library for fast Fourier transforms on real input data", University of Houston, November 2000.

4.3   Conference and journal publications

1. Nadia Shalaby and S. Lennart Johnsson, "A Vector Space Framework for Parallel Stable Permutations", Technical Report TR–32–95, Harvard University.

2. Yu Hu and S. Lennart Johnsson. "Implementing $O(N)$ N–body algorithms efficiently in data parallel languages (High Performance Fortran)". Journal of Scientific Programming, vol. 5, no. 4. pp. 337 – 364, 1996.

3. Yu Hu and S. Lennart Johnsson, "A Data Parallel Implementation of Hierarchical N–body Methods, International Journal of Supercomputer Applications, vol. 10, no. 1, pp. 3 – 40, 1996.

4. Yu Charlie Hu and S. Lennart Johnsson, "High Performance Fortran for Highly Irregular Problems", the 6th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Las Vegas, Nevada, June, 1997.

5. Nadia Shalaby and S. Lennart Johnsson, "Hierarchical Load–Balancing for Parallel Fast Legendre Transforms", the Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, Minnesota, March 14 - 17, 1997.

6. Nadia Shalaby, "Efficient Parallel FFTs for Different Computational Models", the Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, Minnesota, March 14 - 17, 1997.

7. Yu Charlie Hu and S. Lennart Johnsson, "On the Accuracy of Anderson's fast N–body algorithm", the Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, Minnesota, March 14 - 17, 1997.

8. Yu Charlie Hu, Shang–Hua Teng and S. Lennart Johnsson, "A data–parallel implementation of the geometric partitioning algorithm", the Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, Minnesota, March 14 - 17, 1997.

9. Yu Charlie Hu, S. Lennart Johnsson and Shang–Hua Teng, "A data–parallel adaptive N–body method", the Eighth SIAM Conference on Parallel Processing for Scientific Computing – Parallel N-body solver minisymposia, Minneapolis, Minnesota, March 14 - 17, 1997.

10. Yu Hu and S. Lennart Johnsson, "A Data–Parallel Implementation of $O(N)$ Hierarchical N–body Methods", Supercomputing '96, Pittsburg, Philadelphia, PA, November 18 - 22, 1996.

11. Nadia Shalaby and S. Lennart Johnsson, "A Vector–Space Framework for Parallel Stable Permutations", Second International Workshop on Formal Methods for Parallel Programming: Theory and Applications, Geneva, Switzerland, April 1, 1997.

12. Yu Hu, S. Lennart Johnsson, Dimitris Kehagias and Nadia Shalaby, "DPF: A Data parallel Fortran Benchmark Suite", Technical report TR–36–95, Harvard University. Proceedings of the 11th International Parallel Processing Symposium, Geneva, Switzerland, April 1 - 5, 1997.

13. Yu Hu and S. Lennart Johnsson, "Implementing $O(N)$ N–body algorithms efficiently in data parallel languages (High Performance Fortran)". Journal of Scientific Programming, vol. 5, no. 4, pp. 337 – 364, 1996.

14. S. Lennart Johnsson, "Molecular Dynamics Trajectory Databases", NSF Workshop on Interfaces to Scientific Data Archives, Caltech, March 24 – 27, 1998.

15. S. Lennart Johnsson, "Computational Science and Engineering and Technology Exponentials", KDI workshop, Rice University, March 2, 1998.

16. Yu Charlie Hu, S. Lennart Johnsson and Shang–Hua Teng, "Adaptive Order(N) N–body Simulations in HPF", Tutorial on "High Performance Fortran – Practice and Experience", SC97, San Jose, CA, November 15 – 21, 1997

17. Yu Charlie Hu and S. Lennart Johnsson, "Data parallel performance optimizations using array aliasing", In Rob Schreiber, Mike Heath, and Abhiram Ranade, editors, Algorithms for Parallel Processing, IMA Volumes in Mathematics and its Applications. Springer–Verlag, 1997.

18. Y. Charlie Hu, Guohua Jin, S. Lennart Johnsson, Dimitris Kehagias, Nadia Shalaby, "HPFBench: A High Performance Fortran Benchmark Suite", ACM Transaction on Mathematical Software, vol. 26, no. 1, pp. 99 - 149, March, 2000.

19. Dragan Mirković, Rishad Mahasoom, S. Lennart Johnsson, An Adaptive Software Library for Fast Fourier Transforms. Proceedings of the

2000 International Conference on Supercomputing, Santa Fe, NM . pp. 215–224.

References

[1] Dennis M. Healy, Jr., D. Rockmore, Peter J. Kostelec, and Sean S. B. Moore. FFTs for the 2-sphere – improvements and variants. To appear in Advances in Appliad Mathematics.

[2] James R. Driscoll and Jr. Dennis M. Healy. Computing fourier transforms and convolutions on the 2-sphere. *Advances in Appliad Mathematics*, 15:202–250, 1994.

[3] Matteo Frigo and Steven G. Johnson. The fastest fourier transform in the west. Technical Report MIT-LCS-TR-728, MIT, 1997.

[4] V. I. Lebedev. Quadratures on a sphere. *Zh. vychisl. Mat. mat. Fiz.*, 16:293–306, 1976.

[5] Charles Van Loan. *Computational frameworks for the fast Fourier transform*. Philadelphia:SIAM, 1992.

[6] C. M. Rader. Discrete fourier transforms when the number of data samples is prime. *Proceedings of the IEEE*, 56:1107–1108, 1968.

[7] Joseph S. Shang and Datta Gaitonde. Characteristic–based, time–dependent maxwell equation solvers on a general curvilinear frame. *AIAA Journal*, 33:491–498, 1995.

10

User: mirkovic
Host: dhcp12.cs.uh.edu
Class: dhcp12.cs.uh.edu
Job: stdin